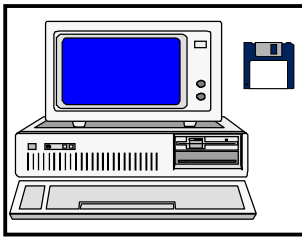


Manip micro N°4 (Vers: 4)
Digicode
(Serrure électronique codée) 1STS

Objectif : réalisation d'une serrure codée à l'aide d'un microcontrôleur Pic.

Présentation du système, cahier des charges.

Le système digicode limite l'accès d'un lieu aux seules personnes en possession du code d'accès. Un code de 4 caractères est saisi puis validé par l'utilisateur sur un clavier situé devant la porte. Si le code saisi correspond au code d'accès stocké en mémoire, la gâche électrique déverrouille la porte pendant 2 secondes environ. Dans le cas contraire une alarme est activée (émission de 4 Bips successifs).
Un afficheur à cristaux liquides signalera à l'utilisateur le nombre de caractères qu'il a déjà entré ; une étoile '*' sera affiché pour chaque caractère.
En cas d'erreur de frappe l'utilisateur pourra recommencer en appuyant sur la touche C (*Clear*).
La validation du code est effectuée par appui sur la touche A (*Acceptation*).

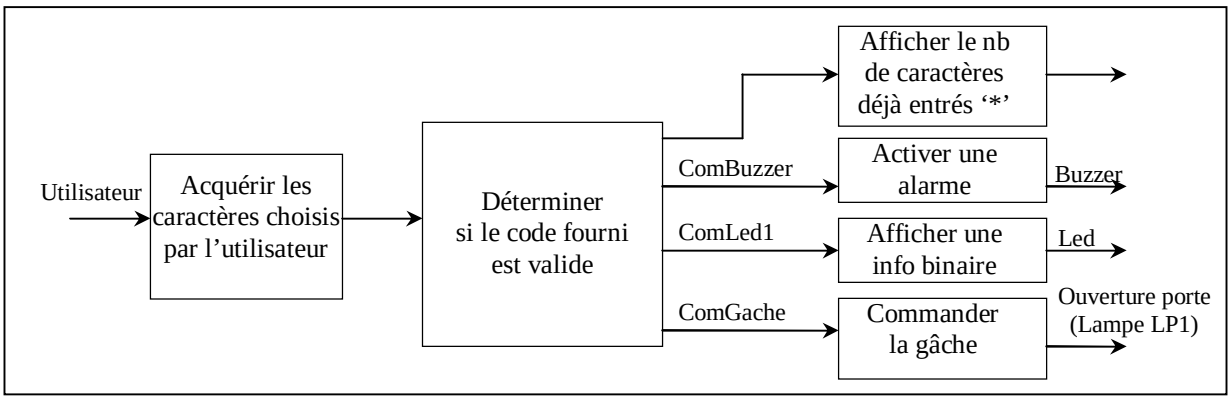
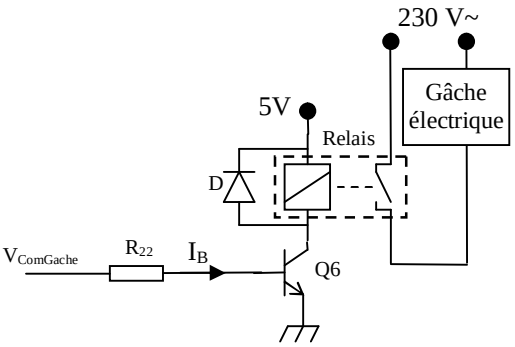


Figure 1 : représentation fonctionnelle du système *digicode*

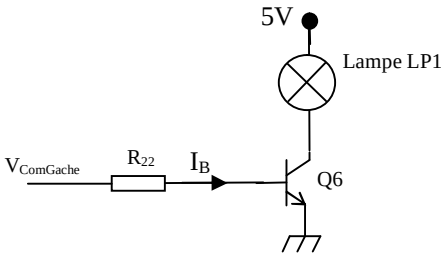
Réalisation à l'aide de la carte Pic Millénium.

- L'acquisition des caractères utilisera le clavier alphanumérique présent sur la carte
- L'affichage utilisera l'écran LCD (2*16 caractères) présent sur la carte
- L'alarme utilisera le buzzer accessible par la broche SPKR (SPeaKeR) du connecteur J16. ATTENTION ; ce buzzer ne comporte pas d'oscillateur intégré. Pour le commander, il est donc nécessaire de générer un signal carré dont la fréquence correspondra à celle du son émis.
- L'état de la gâche sera représenté par la lampe LP1, broche PWM du connecteur J16.

Sur le projet définitif un relais commandera la gâche électrique, voir schéma ci-dessous.



a) structure de la commande de la gâche



b) structure de test de la commande de la gâche

On pourra utiliser des LEDs pour fournir des informations visuelles supplémentaires.

Câblage du microcontrôleur :

- 1) Afficheur LCD et Clavier .

Pour le module d'affichage LCD (compatible: PHILIPS LTN211) et le clavier 16 touches, nous conserverons le câblage réalisé lors de la manip n°3 (Centre d'intérêt: Interface homme machine).

Rappel: **PORTB: RB7 RB6 RB5 RB4 RB3 RB2 RB1 RB0**

col4=PGD col3=PGC col2 col1 Li4 Li3 Li2 Li1

PORTC: RC7 RC6 RC5 RC4 RC3 RC2 RC1 RC0

D7 D6 D5 D4 EN RS --- --- ou RW (RW=0V)

- 2) Câblage du buzzer (borne SPKR), de la LED D1 et de la lampe (borne PWM).

PORTD: RD2 RD1 RD0

PWM D1 SPKR

La lampe de la carte MILLENIUM sera commandée par la broche RD2 à laquelle nous associerons la variable ComGache, en rapport avec son utilisation finale : la commande de la gâche électrique.

Travail à faire

Documents ressources :

- Manipulation N°3 sur l'interface homme machine (Sous programmes de gestion de l'afficheur LCD et du clavier)
- Cours sur le langage C
- Organigrammes fournis dans les pages suivantes.

Vous joindrez à votre compte-rendu un bref manuel d'utilisation du système (doc technique).

I. - Ecriture d'un programme complet pour gérer la serrure avec un code sur 4 chiffres.

Dans cette première version correspondant à l'algorigramme n°1, l'affichage du code ne sera pas masqué par des étoiles. Les symboles correspondant aux touches appuyées par l'utilisateur seront directement affichés (ligne 1, colonne 15) pour faciliter les tests et la mise au point.

I. 1. a) A l'aide de l'algorigramme n°1, écrire le squelette commenté du programme en utilisant les structures génériques suivantes :

```
while (condition)   // commentaire
{
…
}
     ou
do
{
…
} while (condition) ;      // commentaire
```

et

```
if (condition)      // commentaire
{
…
}
else                // commentaire
{
…
}
```

I. 1. b) Complétez le squelette précédent en incluant les appels aux sous-programmes (Utilisez les noms en caractères gras italique comme nom des sous programmes).

Conseils : On utilisera des variables indicateurs : Code_Bon (= 1, pour indiquer si le code entré est correct).

`Char Code_Entre[]=˝----˝;` Permet d'entrer le code pour ouvrir. (ou 4 variables CE1, CE2, CE3 et CE4.

Ex : si `Code_Entre[0]`= 4 , `Code_Entre[1]`= 5 , `Code_Entre[2]`= 2 , `Code_Entre[3]`= 3 . Alors le code sera « 4523 ». (idem avec : CE1=4 CE2=5, etc…).

Rem : le dernier caractère entré est donc le 3. Le premier caractère « le 4 » passera de la position 3 à la position 0 (décalage). => « ---4 » puis « --45 » puis « -452 » et enfin « 4523 ».
Pour le code secret, on utilisera de même soit un tableau : const char Code_Secret[]=˝4523˝;
Soit 4 variables CS1= 4, CS2= 5 etc …

I. 2. A l'aide de l'algorigramme n°2, écrire le squelette commenté du sous-programme Clavi.
I. 3. Modifiez la fonction Scrut_Clav écrit lors de la manip n°3, pour en faire un sous-programme ou la variable touche sera globale et le rendre conforme à l'algorigramme n°3 (Ajout de la variable Nbtou pour compter le nombre de touches enfoncées).

I. 4. Ecrire les sous-programmes restant.

II. - Ecriture d'une deuxième version du programme. (Ne pas perdre la 1ere version)

Il s'agit maintenant de modifier l'affichage pour le rendre conforme au cahier des charges.
Nous procèderons en trois étapes.

1. Modifiez le squelette du programme pour le rendre conforme à l'algorigramme n°4.
2. Ecrire ***Affiche_Cod*** de telle sorte que ce sous-programme affiche les caractères correspondants au code entré par l'utilisateur (colonne 10, 11, 12 et 13 de la ligne 1).
3. Modification de ***Affiche_Cod***. Pour respecter le cahier des charges sans perdre le confort qu'apportait l'affichage des touches appuyées, vous utiliserez une constante binaire DEBUG dont la valeur associée à la directive #if nous permettra de faire fonctionner le sous programme d'affichage dans les deux modes suivants :

DEBUG=1 ; fonctionnement en mode debug (identique au II. 2) ;
DEBUG=0 ; fonctionnement conforme au cahier des charges (affichage masqué) : une étoile '*' pour chaque caractère entré et un '-' pour les caractères restant à taper.

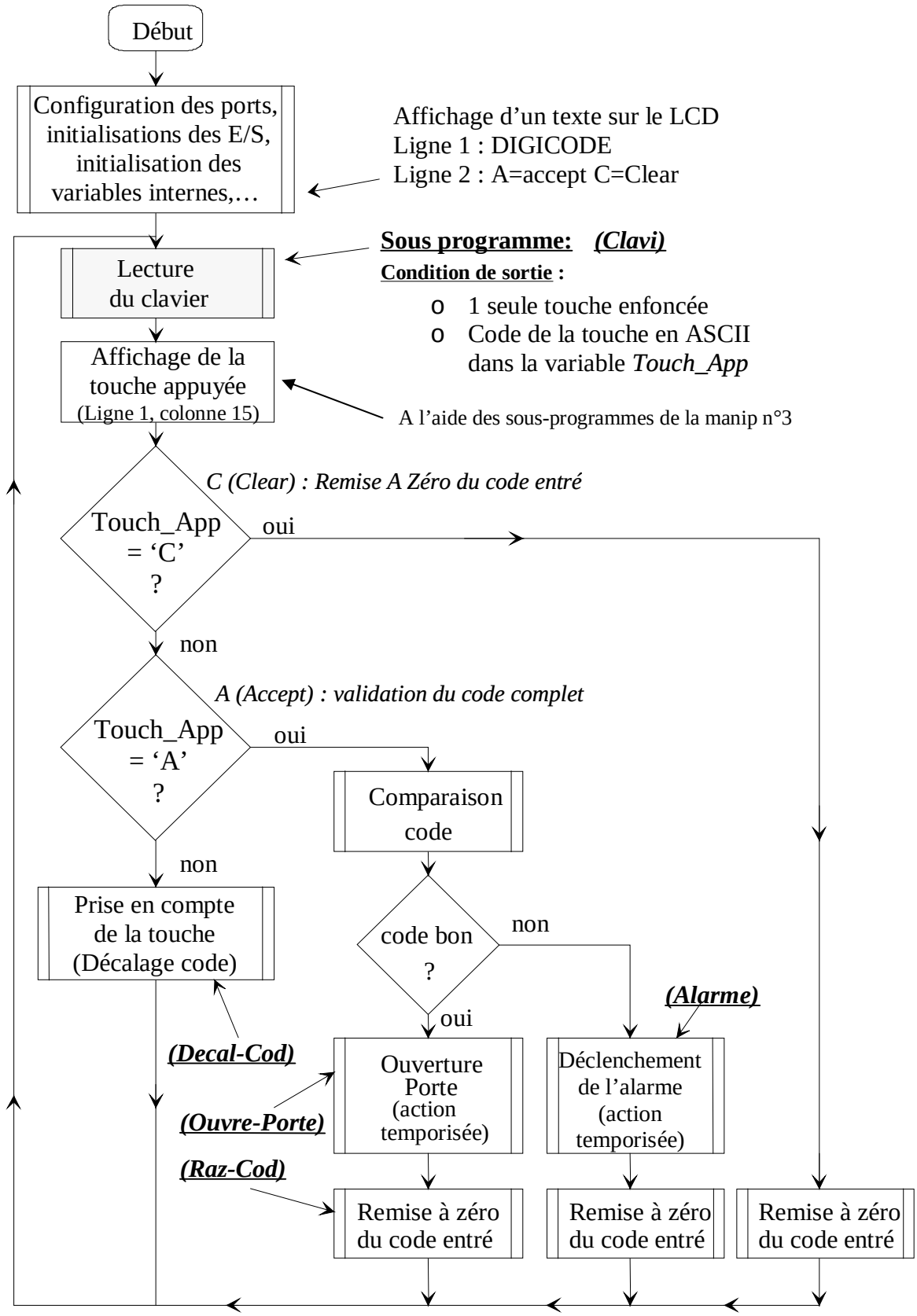
Evolutions et exploitations possible du programme.

La structure du programme et le type d'interfaces utilisées répondent à des besoins très génériques. En effet il existe un grand nombre de systèmes dont la finalité est très différente du digicode mais dont le matériel et la gestion associée sont par contre très semblables.

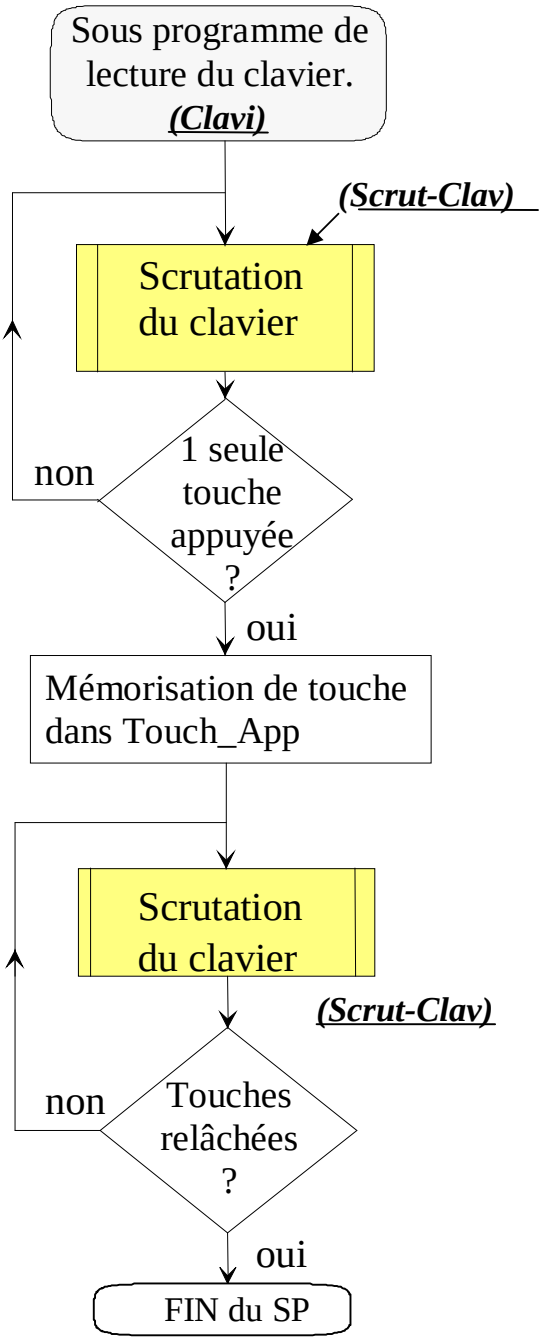
L'un des points fort des micro-contrôleurs :
A l'aide d'une platine de test, on peut développer un grand nombre de projets sans avoir besoin de concevoir de circuits imprimés.
Une fois en possession d'une bibliothèque de sous-programmes, la partie traitement d'une quantité de projets peuvent être très rapidement développée.
Ajoutée au faible coût matériel et logiciel des outils de développement et des composants eux-mêmes.

Le microcontrôleur s'est imposé dans un très vaste éventail d'application de l'électronique.

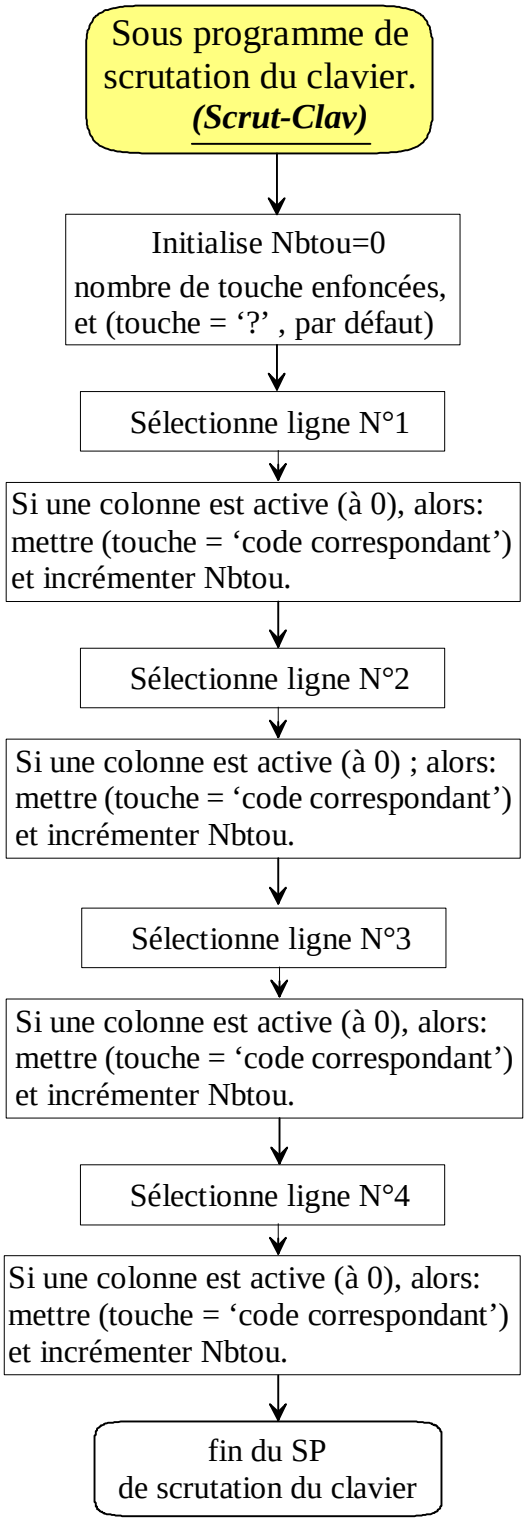
Algorigramme n°1 : Proposition d'algorigramme de la serrure codée


Début
Configuration des ports, initialisations des E/S, initialisation des variables internes,…
Affichage d'un texte sur le LCD
Ligne 1 : DIGICODE
Ligne 2 : A=accept C=Clear
Sous programme: (Clavi)
Condition de sortie :
o 1 seule touche enfoncée
o Code de la touche en ASCII dans la variable Touch_App
Lecture du clavier
Affichage de la touche appuyée (Ligne 1, colonne 15)
A l'aide des sous-programmes de la manip n°3
C (Clear) : Remise A Zéro du code entré
Touch_App = 'C' ?
oui
non
A (Accept) : validation du code complet
Touch_App = 'A' ?
oui
Comparaison code
non
Prise en compte de la touche (Décalage code)
code bon ?
non
(Alarme)
oui
(Decal-Cod)
(Ouvre-Porte)
(Raz-Cod)
Ouverture Porte (action temporisée)
Déclenchement de l'alarme (action temporisée)
Remise à zéro du code entré
Remise à zéro du code entré
Remise à zéro du code entré

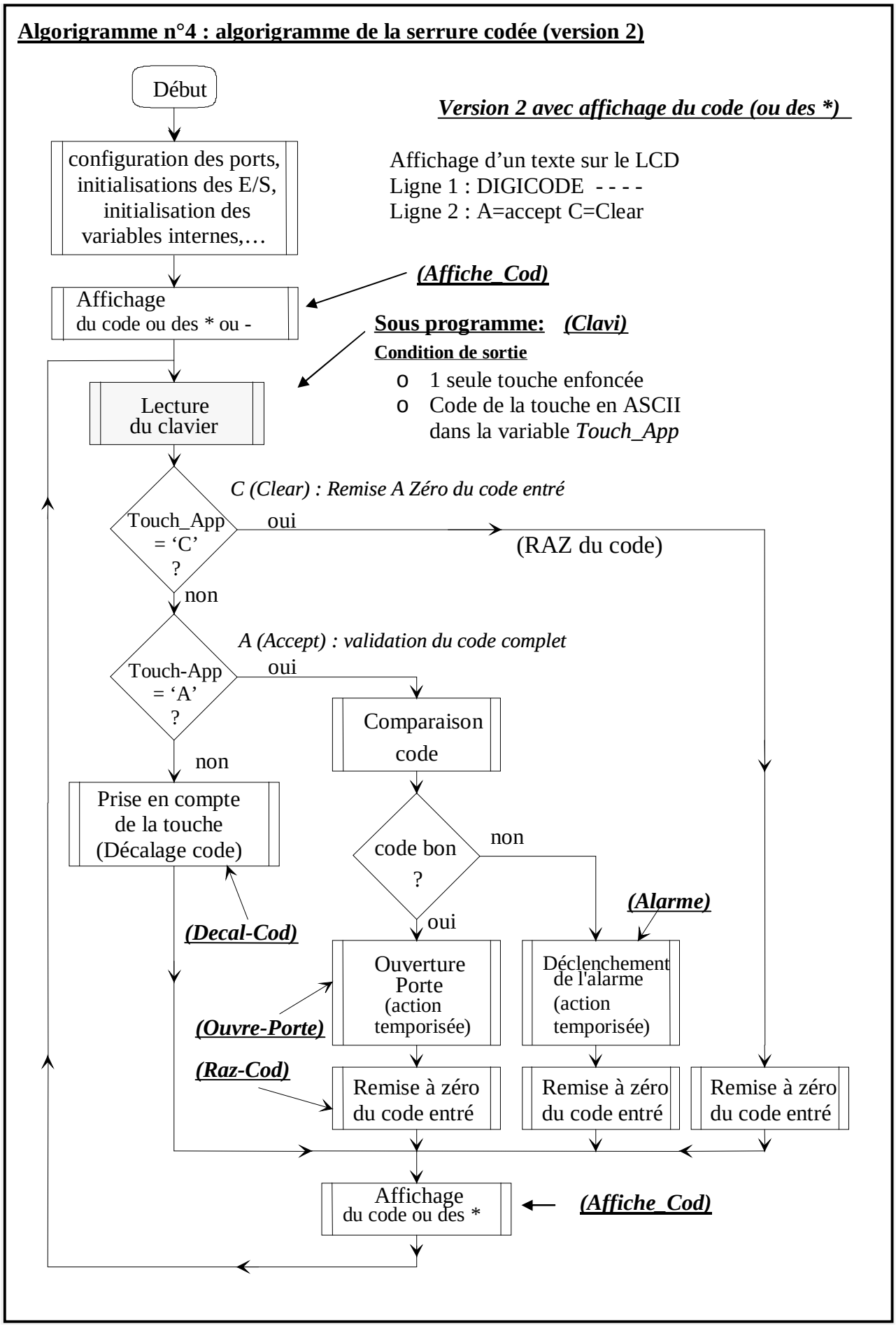
Algorigramme n°2 : sous programme *Clavi* .



Algorigramme n°3 : sous-prog. *Scrut_Clav* .


Sous programme de scrutation du clavier.
(Scrut-Clav)
Initialise Nbtou=0
nombre de touche enfoncées, et (touche = '?' , par défaut)
Sélectionne ligne N°1
Si une colonne est active (à 0), alors: mettre (touche = 'code correspondant') et incrémenter Nbtou.
Sélectionne ligne N°2
Si une colonne est active (à 0) ; alors: mettre (touche = 'code correspondant') et incrémenter Nbtou.
Sélectionne ligne N°3
Si une colonne est active (à 0), alors: mettre (touche = 'code correspondant') et incrémenter Nbtou.
Sélectionne ligne N°4
Si une colonne est active (à 0), alors: mettre (touche = 'code correspondant') et incrémenter Nbtou.
fin du SP de scrutation du clavier

Algorigramme n°4 : algorigramme de la serrure codée (version 2)


Début
configuration des ports, initialisations des E/S, initialisation des variables internes,…
Affichage du code ou des * ou -
(Affiche_Cod)
Version 2 avec affichage du code (ou des *)
Affichage d'un texte sur le LCD
Ligne 1 : DIGICODE - - - -
Ligne 2 : A=accept C=Clear
Sous programme: (Clavi)
Condition de sortie
o 1 seule touche enfoncée
o Code de la touche en ASCII dans la variable Touch_App
Lecture du clavier
C (Clear) : Remise A Zéro du code entré
Touch_App = 'C' ?
oui
(RAZ du code)
non
A (Accept) : validation du code complet
Touch-App = 'A' ?
oui
Comparaison code
non
Prise en compte de la touche (Décalage code)
code bon ?
non
oui
(Alarme)
(Decal-Cod)
Ouverture Porte (action temporisée)
Déclenchement de l'alarme (action temporisée)
(Ouvre-Porte)
(Raz-Cod)
Remise à zéro du code entré
Remise à zéro du code entré
Remise à zéro du code entré
Affichage du code ou des *
(Affiche_Cod)